プログラム規模を考慮した離散ワイブル型ソフトウェア信頼性モデル(不確実性を含む意思決定の数理とその応用)

AUTHOR(S):

井上, 真二; 山田, 茂

数理解析研究所講究録
第 1548 巻 2007 年 85-90

プログラム規模を考慮した離散ワイブル型ソフトウェア信頼性モデル

鳥取大学・工学部　井上 真二 (Shinji Inoue)†
鳥取大学・工学部　山田　茂 (Shigeru Yamada)†
†Faculty of Engineering, Tottori University

1　はじめに

ソフトウェア開発工程のテスト工程では，要求仕様定義，設計，コーディングを経て製造されたソフトウェアシステムの最終的な品質／信頼性の確認作業が行われる．テスト工程におけるソフトウェアの実行過程に沿って観測されるソフトウェアの信頼度成長現象は，一般的に，テスト作業に費やした労力やテストケース設計者の能力など様々な要因が影響していることが知られている [1, 2]．さらに，サイクロマチック数などのメトリクスに代表されるソフトウェアの複雑性 [3] も，テスト工程におけるソフトウェア信頼度成長現象に大きく影響を及ぼしている要因の１つとして考えることができる．したがって，ソフトウェア信頼度成長過程とソフトウェア複雑性メトリクスとの関連づけを行いながら，信頼性評価を行うことは重要な問題である．

本研究では，実際のソフトウェアプロジェクトにおいて容易に計測可能なソフトウェア複雑性メトリクスとして，プログラム規模（コード行数）を取り挙げ，ソフトウェア故障発生時間の順序統計量に着目して構築されるソフトウェア信頼性モデル（SRM：software reliability model）の一般化枠組み [4, 5] に基づいて，プログラム規模を考慮した離散型 SRM の構築枠組みに関する議論を行う．実際のテスト工程では，ソフトウェア故障発生現象やフォールト発見事象に関するデータを採取するために，連続的にテスト期間中のソフトウェアの挙動を監視する場合もあるが，多くの場合，ある一定の時間区間ごとのソフトウェア故障発生頻度もしくは修正・除去されたフォールト数を計測する．このような場合，離散型 SRM は，比較的に整合性を有するモデルとして考えられる．また，実行されたテストケース数などをテスト時間として取り扱う場合にも，同様のことが言える．本研究では，さらに，2 種類のソフトウェア故障発生時間分布を考え，議論した離散型 SRM の構築枠組みに基づいた新たな離散型 SRM を構築する．

2　一般化枠組み

今回議論する離散型 SRM 構築枠組みは，テスト工程におけるソフトウェア故障発生現象に対して，まず，以下のような基本的仮定 [4, 5] を設ける．

(A1)　ソフトウェア故障が発生した場合，その原因となるフォールトは，直ちにかつ完全に修正・除去される．

(A2)　各ソフトウェア故障は，それぞれ，独立かつ時間に関してランダムに発生し，各ソフトウェア故障発生時刻は，それぞれ，同一の離散型確率分布 $P(i) \equiv \Pr\{I \leq i\} = \sum_{k=0}^{i} p_I(k)$ $(i = 0, 1, 2, \cdots)$ に従う．ここで，$p_I(k)$ および $\Pr\{A\}$ は，それぞれ，I に関する確率関数および事象 A に対する確率を表す．

(A3)　テスト開始前にソフトウェア内に潜在する総フォールト数（初期潜在フォールト数）N_0 は，ある確率分布に従う確率変数とする．

いま，テスト開始後 i 期目までに発見される総フォールト数を表す離散型確率過程 $\{N(i),\ i = 0, 1, \cdots\}$ を導入する．このとき，テスト開始後 i 期目までに m 個のフォールトが発見される確率関数は，ソフトウェア故障発生現象もしくはフォールト発見事象に関する上述の基本的仮定から，

$$\Pr\{N(i) = m\} = \sum_{n} \binom{n}{m} \{P(i)\}^m \{1 - P(i)\}^{n-m} \Pr\{N_0 = n\} \qquad (m = 0, 1, 2, \cdots, n), \tag{1}$$

のように求められる．

本研究では，式 (1) における初期潜在フォールト数を表す確率変数 N_0 に対して，以下のような，パラメータ (K,λ) をもつ二項分布を仮定することで，プログラム規模を考慮した離散型 SRM を構築するための枠組みを与える．

$$\Pr\{N_0=n\}=\binom{K}{n}\lambda^n(1-\lambda)^{K-n} \qquad (0<\lambda<1\,;\, n=0,1,\cdots,K). \tag{2}$$

式 (2) には，初期潜在フォールト数に関する以下のような物理的意味が含まれている．

(B1) テスト開始時点におけるプログラムは，K コード行 (LOC) で構成される．

(B2) 各コードは，それぞれ一定の確率 λ で 1 個のフォールトを含む．

(B3) プログラム内のコード中に潜在するフォールトにより引き起こされるソフトウェア故障は，それぞれ独立かつランダムに発生する．

初期潜在フォールト数を表す確率変数 N_0 に，式 (2) のような二項分布を仮定することによって，プログラム規模がソフトウェア信頼度成長過程に与える影響を，SRM へ反映することができる [6]．式 (2) を式 (1) に代入して整理すると，テスト開始後 i 期目までに m 個のフォールトが発見される確率関数は，次式のように求められる．

$$\Pr\{N_B(i)=m\}=\binom{K}{m}\{\lambda P(i)\}^m\{1-\lambda P(i)\}^{K-m} \qquad (m=0,1,2,\cdots K). \tag{3}$$

3 ソフトウェア信頼性評価尺度

ソフトウェア信頼性評価尺度は，ソフトウェアの信頼性を定量的に計測・評価するために有用な尺度である．本研究では，式 (1) に示した基本的仮定に基づいて，良く知られている代表的な信頼性評価尺度を一般的に記述する．

3.1 発見フォールト数の期待値および分散

式 (1) より，テスト開始後 i 期目までに発見される総フォールト数 $N(i)$ の期待値 $\mathrm{E}[N(i)]$ は，次のように導出される．

$$\begin{aligned}\mathrm{E}[N(i)]&=\sum_{z=0}^{n}z\sum_{n}\binom{n}{z}\{P(i)\}^z\{1-P(i)\}^{n-z}\Pr\{N_0=n\}\\&=\mathrm{E}[N_0]P(i).\end{aligned} \tag{4}$$

また，同様に，$N(i)$ の分散 $\mathrm{Var}[N(i)]$ は，

$$\begin{aligned}\mathrm{Var}[N(i)]&=\mathrm{E}[N(i)^2]-(\mathrm{E}[N(i)])^2\\&=\mathrm{Var}[N_0]\{P(i)\}^2+\mathrm{E}[N_0]P(i)\{1-P(i)\},\end{aligned} \tag{5}$$

のように導出できる．したがって，初期潜在フォールト数を表す確率変数 N_0 が，式 (2) のような二項分布に従う場合，テスト開始後 i 期目までに発見される総フォールト数 $N_B(i)$ の期待値および分散は，式 (4) および式 (5) から，それぞれ，

$$\mathrm{E}[N_B(i)]=K\lambda P(i), \tag{6}$$

$$\mathrm{Var}[N_B(i)]=K\lambda P(i)\{1-\lambda P(i)\}, \tag{7}$$

のように求められる．このとき，$K\lambda$ は，テスト開始前にソフトウェア内に潜在する総期待フォールト数を表す．

3.2 ソフトウェア信頼度関数

ソフトウェア信頼度は，有用なソフトウェア信頼性評価尺度の1つとして知られている．離散時間に依存するソフトウェア信頼度成長過程を取り扱う場合，ソフトウェア信頼度は，テスト開始後i期目までテストが進行しているとき，時間区間$(i, i+h](i, h = 0, 1, \cdots)$においてソフトウェア故障が発生しない確率として定義される[7]．これより，離散型ソフトウェア信頼度関数$R(i, i+h)$は，式(1)から，

$$\begin{aligned} R(i,h) &= \sum_k \Pr\{N(i+h) = k \mid N(i) = k\} \cdot \Pr\{N(i) = k\} \\ &= \sum_k \left[\{P(i)\}^k \{1 - P(i+h)\}^{-k} \sum_n \binom{n}{k} \{1 - P(i+h)\}^n \cdot \Pr\{N_0 = n\}\right], \end{aligned} \tag{8}$$

のように導出される．式(8)より，初期潜在フォールト数N_0が，式(2)の二項分布に従う場合，離散型ソフトウェア信頼度関数$R_B(i, h)$は，

$$R_B(i,h) = [1 - \lambda\{P(i+h) - P(i)\}]^K, \tag{9}$$

のように導出できる．

4 ソフトウェア故障発生時間分布

式(3)に示したモデリング枠組みに基づいて，プログラム規模を考慮した離散型SRMを構築するためには，式(3)に含まれる離散型ソフトウェア故障発生時間分布$P(i)$に対して，適切な離散型確率分布を適用する必要がある．本研究では，2種類の離散型ソフトウェア故障発生時間分布を構築して，離散ワイブル型SRMおよび離散テスト網羅度依存型SRMをそれぞれ提案する．

4.1 離散型ワイブル分布

本研究では，式(3)のソフトウェア故障発生時間分布$P(i)$に対して，離散型ワイブル分布（discrete Weibull distribution）[8]を適用することにする．離散型ワイブル分布の分布関数は，次のように与えられる．

$$P(i) = 1 - (1-p)^{i^\beta} \qquad (i = 0, 1, 2, \cdots;\ \beta > 0,\ 0 < p < 1). \tag{10}$$

ここで，pは単位期間当りに1つのソフトウェア故障が発生する確率を表し，βは形状パラメータである．また，1つのソフトウェア故障に対するハザードレートは，式(10)から，

$$z(i) = 1 - (1-p)^{(i+1)^\beta - i^\beta}, \tag{11}$$

と導出できる．すなわち，式(10)の離散型ワイブル分布は，形状パラメータβによって，ソフトウェア故障発生時間分布の確率的挙動を柔軟に記述することができる．式(3)に示したモデリング枠組みにおいて，式(10)の離散型ワイブル分布を適用したときに構築される離散型SRMを離散ワイブル型SRMと名づける．

4.2 テスト網羅度を考慮した離散型確率分布

テスト工程におけるテスト網羅度達成状況の時間的推移を考慮した離散型ソフトウェア故障発生時間分布を考える．ここで，離散型ソフトウェア故障発生時間分布に対するハザードレートを，テスト開始後$(i-1)$期目までにフォールトが検出がされない条件の下で，その後のi期目においてフォールトが検出される確率として定義する．このとき，離散型ハザードレート$z(i)$は，

$$z(i) \equiv \Pr\{I = i \mid I > i-1\} = \frac{P(i) - P(i-1)}{1 - P(i-1)} \tag{12}$$

と表現できる．式(12)より，離散型ソフトウェア故障発生時間分布$P(i)$は，以下のように求められる．

$$\left.\begin{aligned} &P(0) = 0 \\ &P(i) = 1 - \prod_{j=1}^{i} (1 - z(j)) \quad (i = 1, 2, \cdots) \end{aligned}\right\}. \tag{13}$$

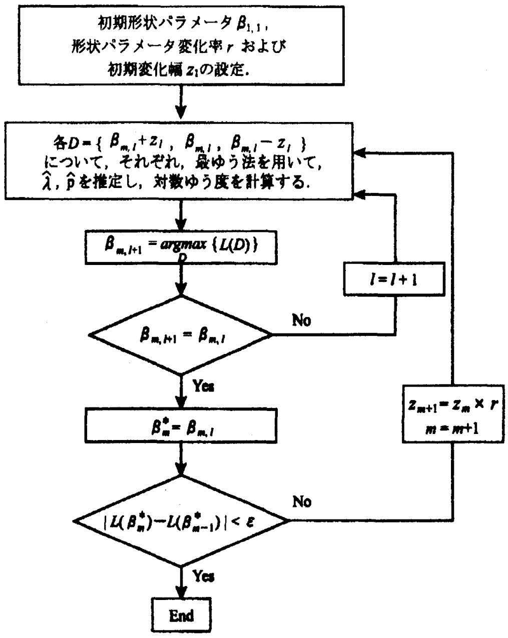

図 1： 離散ワイブル型 SRM に対するパラメータ推定アルゴリズム.

式 (13) より，離散型ソフトウェア故障発生時間分布を与えることは，それに対応した離散型ハザードレートを与えることに帰着する.

本研究では，式 (12) の離散型ハザードレートを次式のように与える.

$$z(i) = \phi(i)C(i). \tag{14}$$

式 (14) において，$\phi(i)$ はテスト開始後 i 期目における単位テスト網羅度当りのハザードレートであり，$C(i)$ はテスト開始後 i 期目までに達成されたテスト網羅度を表す関数である．特に，$C(i)$ をテスト網羅度関数 (testing-coverage function) と呼ぶこととする.

式 (14) の離散型ハザードレートを具体的に特徴付けるにあたり，本研究では，$\phi(i) \equiv \phi$ として，以下に示すテスト網羅度関数を導入する.

$$C(i) = \frac{\alpha\left[1 - \left\{(1-\frac{1}{2}b)/(1+\frac{1}{2}b)\right\}^i\right]}{1 + z\left\{(1-\frac{1}{2}b)/(1+\frac{1}{2}b)\right\}^i} \qquad (0 < \alpha < 1,\ z > 0,\ b > 0). \tag{15}$$

ここで，α はテスト終了時におけるテスト網羅度の達成目標値，b は単位テスト期数当りのテスト網羅度達成率，z はテストケース設計者の設計能力を表す．式 (15) は，テストケース設計者の設計能力を考慮した連続型テスト網羅度関数 [2] の基本的仮定から導出される微分方程式を，双線形化法 [9] と呼ばれる差分化手法を用いて得られる差分方程式の厳密解である．テスト網羅度を考慮した離散型ソフトウェア故障発生時間分布を適用したときに構築される離散型 SRM を離散テスト網羅度依存型 SRM と名づける.

5 パラメータ推定

ソフトウェア故障発生時間分布に式 (10) の離散型ワイブル分布を適用した場合について議論する．本研究で提案する離散ワイブル型 SRM について，通常よく使われる最ゆう法を単純に適用した場合，パラメータ λ, p, および β を同時に推定することが極めて困難である．そのため，本研究では，ゆう度に基づいた発見的アルゴリズム [10] に基づいて，提案した離散ワイブル型 SRM のパラメータ推定を行う.

図 1 に，提案した離散ワイブル型 SRM のパラメータ推定アルゴリズムを示す．ここに示すパラメータ推定アルゴリズムは，ます最初に，離散ワイブル型 SRM に含まれる形状パラメータ β を事前に設定した変化

表 1： MSE に基づいたモデルの適合性比較結果.

	Model 1	Model 2	D-GOMP	D-LOGI	GEDR
DS1	**35.023**	248.62	70.344	346.78	603.20
DS2	**25.979**	112.55	744.89	1103.6	73.089

幅に応じて固定した上で，パラメータ p および λ を最ゆう法により推定を行った後，その結果に基づいて，パラメータ β の推定を行うものである．

ある固定された形状パラメータ β に対して，パラメータ p および λ を推定するための対数ゆう度関数は，次のように求められる．はじめに，一定のテスト時間間隔 $(0,\ t_k]$ において発見された総フォールト数 y_k に関する N 組のフォールト発見数データ $(t_k,\ y_k)(k = 0, 1, 2, \cdots, N)$ が観測されたものとする．まず，$\{N_B(i), i = 0, 1, 2, \cdots\}$ に関するゆう度関数 l は，ベイズの定理および確率過程 $\{N_B(i), i = 0, 1, 2, \cdots\}$ が有するマルコフ性を用いて，

$$
\begin{aligned}
l &\equiv \Pr\{N_B(t_1) = y_1, N_B(t_2) = y_2, \cdots, N_B(t_N) = y_N\} \\
&= \prod_{i=2}^{N} \Pr\{N_B(t_i) = y_i \mid N_B(t_{i-1}) = y_{i-1}\} \Pr\{N_B(t_1) = y_1\}, \qquad (16)
\end{aligned}
$$

のように導出できる．ここで，式 (16) に含まれる条件付確率 $\Pr\{N_B(t_i) = y_i \mid N_B(t_{i-1}) = y_{i-1}\}$ は，次のように書き換えることができる．

$$
\Pr\{N_B(t_i) = y_i \mid N_B(t_{i-1}) = y_{i-1}\} = \binom{K - y_{i-1}}{y_i - y_{i-1}} \{z(t_{i-1}, t_i)\}^{y_i - y_{i-1}} \{1 - z(t_{i-1}, t_i)\}^{K - y_i}. \qquad (17)
$$

ここで，

$$
z(t_{i-1}, t_i) = \frac{\lambda\{P(t_i) - P(t_{i-1})\}}{1 - \lambda P(t_{i-1})}, \qquad (18)
$$

である．式 (17) を式 (16) に代入して整理すると，最終的に，ゆう度関数 l は，

$$
l = \prod_{i=1}^{N} \binom{K - y_{i-1}}{y_i - y_{i-1}} \{z(t_{i-1}, t_i)\}^{y_i - y_{i-1}} \{1 - z(t_{i-1}, t_i)\}^{K - y_i}; \qquad (19)
$$

のように求められる．ただし，$t_0 = 0$, $y_0 = 0$, and $P(t_0) = 0$ である．したがって，対数ゆう度関数 L は，式 (19) の両辺に自然対数をとることで，

$$
\begin{aligned}
\log l &\equiv L \\
&= \log K! - \log\{(K - y_N)!\} - \sum_{i=1}^{N} \log\{(y_i - y_{i-1})!\} + y_N \log \lambda \\
&\quad + \sum_{i=1}^{N} (y_i - y_{i-1}) \log\{P(t_i) - P(t_{i-1})\} + (K - y_N) \log\{1 - \lambda P(t_N)\}, \qquad (20)
\end{aligned}
$$

のように導出できる．形状パラメータ β は所与のため，式 (20) から，パラメータ p および λ に関する同時対数ゆう度方程式を数値的に解くことで，パラメータ p および λ の推定値 $\widehat{p}$ および $\widehat{\lambda}$ を，それぞれ得ることができる．

離散テスト網羅度依存型 SRM に関するパラメータ推定手法については，テスト工程におけるテスト網羅度達成状況を示した実測データからテスト網羅度関数に含まれるパラメータを最小 2 乗法を用いて推定し [2]，その後，テスト網羅度達成状況に対応したフォールト発見数データに基づき，式 (20) の対数ゆう度関数を用いてパラメータ (λ, ϕ) を最ゆう法により推定するような 2 段階のパラメータ推定手順をとる．

6 モデルの適合性比較

実際のテスト工程において得られたテスト網羅度データおよびそれに対応したフォールト発見数データを用いて，今回提案した離散ワイブル型 SRM（Model1）および離散テスト網羅度依存型 SRM（Model2）の 2 組の離散型 SRM と既存の離散型 SRM との適合性比較を行う．本研究において取り上げる既存の離散型 SRM は，離散型ゴンペルツ曲線モデル（D-GOMP）[11]，離散型ロジスティック曲線モデル（D-LOGI）[11]，および幾何減少型フォールト発見率モデル（GEDR）[7] である．また，本研究において用いる 2 つの実測データ [12] は，DS1 および DS2 と名づけることにする．

表 1 に平均偏差平方和（MSE）に基づいたモデルの適合性比較結果を示す．表 1 より，今回提案した離散ワイブル型 SRM は，テスト工程のおけるソフトウェア信頼度成長過程を最も精度良く推定できていることがわかる．また，離散テスト網羅度依存型 SRM は，適合性比較の結果，今回取り上げた既存の離散型 SRM よりも精度良く推定できることは確認できなかった．

7 おわりに

本研究では，プログラム規模を考慮した離散型 SRM の構築枠組みの下で，2 種類のソフトウェア故障発生時間分布を考え，離散ワイブル型 SRM および離散テスト網羅度依存型 SRM をそれぞれ構築した．実測データを用いた既存の離散型 SRM との適合性比較結果では，離散ワイブル型 SRM が今回取り上げた離散型 SRM の中で最もよく実測データに対して適合していることがわかった．一方，離散テスト網羅度依存型 SRM は，既存の離散型 SRM よりも実測データに対する適合性が向上していることが確認できなかった．今後は，離散型テスト網羅度依存型 SRM について，モデルの改良等によりモデルが有する適合性の向上を図ると共に，より多くの実測データを用いて，提案モデルの有効性および妥当性を検証する必要がある．

謝辞

本研究の一部は，日本学術振興会科学研究費補助金 基盤研究 (C)（課題番号 18510124）の援助を受けたことを付記する．

参考文献

[1] S. Yamada, J. Hishitani, and S. Osaki, "Software-reliability growth with a Weibull test-effort: A model & application," *IEEE Trans. Rel.*, vol. 42, no. 1, pp. 100–106, 1993.

[2] S. Inoue and S. Yamada, "Testing-coverage dependent software reliability growth modeling," *Int. J. Relib. Qual. Saf. Eng.*, vol. 11, no. 4, pp. 303–312, 2004.

[3] 山田茂，高橋宗雄，ソフトウェアマネジメントモデル入門，共立出版，東京，1993.

[4] D.R. Miller, "Exponential order statistic models of software reliability growth," *IEEE. Trans. Softw. Eng.*, vol. SE-12, no. 1, pp. 12–24, 1986.

[5] H. Okamura, M. Ando, and T. Dohi, "Generalized-gamma software reliability model," *Trans. IEICE*, vol. J87-D-I, no. 8, pp. 805–814, 2004.

[6] M. Kimura, S. Yamada, H. Tanaka, and S. Osaki, "Software reliability measurement with prior-information on initial fault content," *Trans. IPS Japan*, vol. 34, no. 7, pp. 1601–1609, 1993.

[7] S. Yamada and S. Osaki, "Discrete software reliability growth models," *Appli. Stoc. Mod. Data Ana.*, vol. 1, no. 1, pp. 65–77, 1985.

[8] T. Nakagawa and S. Osaki, "The discrete Weibull distribution," *IEEE Trans. Rel.*, vol. R-24, no. 5, pp. 300–301, 1975.

[9] R. Hirota, "Nonlinear partial difference equations. V. Nonlinear equations reducible to linear equations," *J. Physic. Soc. Japan*, vol. 46, no. 1, pp. 312–319, 1979.

[10] 岡村寛之，安藤光昭，土肥　正，"一般化ガンマソフトウェア信頼性モデル，" 電子情報通信学会論文誌，vol. J87-D-I, no. 8，pp. 805–814，2004 年 8 月.

[11] D. Satoh and S. Yamada, "Discrete equations and software reliability growth models," *Proc. 12th IEEE Int. Symp. Softw. Reliab. Eng.* (*ISSRE'01*), pp. 176–184, 2001.

[12] T. Fujiwara and S. Yamada, "A new testing-path coverage measure — Testing-domain metrics based on a software reliability growth model —," *Proc. 13th IEEE Int. Symp. Softw. Reliab. Eng.* (*ISSRE'02*), 2002, pp. 71–75.